UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/982,243	10/16/2001	Ashish Prakash	67272-8048.US02	8578

77042 7590 01/22/2009

Perkins Coie LLP
P.O. Box 1208
Seattle, WA 98111-1208

EXAMINER	
CHOUDHURY, AZIZUL Q	

ART UNIT	PAPER NUMBER
2445	

MAIL DATE	DELIVERY MODE
01/22/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/982,243	PRAKASH ET AL.	
	Examiner	Art Unit	
	AZIZUL CHOUDHURY	2445	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 October 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *21,23,25-34,36,38,39 and 46-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *21, 23, 25-34, 36, 38-39 and 46-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 February 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

Detailed Action

This office action is in response to the correspondence received on October 9, 2008.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 21, 23, 25-34, 36, 38-39 and 46-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Abraham et al (US Patent No: US005983270A) in view of Hyder et al (US Patent No: 5,983,274), hereafter referred to as Abraham and Hyder, respectively.

1. With regards to claim 21, Abraham teaches through Hyder, a device to log information in a network cache, the device comprising: an application module *(equivalent to network management program; see column 2, lines 30-32, Abraham)* to receive requests from a plurality of clients *(equivalent to devices within the intranet and see element 44 within Figure 2, Abraham)* for content maintained by an origin server *(equivalent to devices within the internet and see element 40 within Figure 2, Abraham)* and to receive responses to the requests from the origin server *(inherent feature in a network yet is also taught within column 2, lines 31-53, Abraham)*; an interface to allow selection of a protocol,

including a selection for logging of some or all of a plurality of fields of the protocol that may be present in each of said requests and responses *(see column 14, lines 9-28, Abraham)*, a first data structure to store, for each of the plurality of fields of the protocol, an indication of whether a field has been selected for logging, wherein when the field has been selected for logging, the indication identifies a position in the specified sequence of the selected field (*see Hyder below*); a second data structure to store information corresponding to each selected field (*see Hyder below*); a third data structure to store, for each of the selected fields, a reference to the corresponding information stored in the second data structure, wherein each reference is stored in a location of the third data structure that corresponds to the position in the specified sequence of the selected field, and wherein the position is identified by the indication stored in the first data structure that corresponds to the selected field (*see Hyder below*); and specification of a sequence in which the selected fields are to appear in a log file *(see column 7, lines 29-31, Abraham)*; and a log module to record information of the received requests and response into the log file, according to the selected protocol, the selected fields of the protocol, and the specified sequence *(see column 15, lines 34-40, Abraham)*.

While Abraham teaches a network management system capable of logging the network data however, Abraham fails to explicitly cite the claimed data structures. In particular, Abraham does not explicitly cite a first data structure to store a value indicating a position in the specified sequence for each

selected field; a second data structure to store information corresponding to each selected field; and a third data structure to store a reference to the information stored in the second data structure, including storing the reference in a location of the third data structure that corresponds to the position in the specified sequence of a field corresponding to the reference. In the same field of endeavor, Hyder teaches a system for maintaining network information. In particular, Hyder teaches a control data structure and packet data structure *(see Figure 4 and column 8, line 35 - column 9, line 12, Hyder).* The control data structure has within it a storage section (data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see column 9, line 2 and Figure 4, Hyder)*. The control data structure also has within it a storage section (data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Finally Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure (deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*. The data structures of Hyder's design allow the system to maintain network information (log network data). Therefore it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Abraham with those of Hyder, to teach how data structures can be associated with network information *(see column 5, lines 13-15, Hyder)*.

2. With regards to claim 23, Abraham teaches through Hyder, the device wherein the interface allows the creation of new fields in addition to the plurality of fields *(see element 230, Figure 7A, Abraham)*.

3. With regards to claim 25, Abraham teaches through Hyder, the device wherein the interface is a graphical user interface *(see column 14, line 11, Abraham)*.

4. With regards to claim 26, Abraham teaches through Hyder, the device wherein the interface is a command line interface *(equivalent to low level commands; see column 53, lines 15-23, Abraham).*

5. With regards to claim 27, Abraham teaches through Hyder, a method of logging information in a network cache, the method comprising: receiving requests from a plurality of clients *(equivalent to devices within the intranet and see element 44 within Figure 2, Abraham)* for content maintained by an origin server *(equivalent to devices within the internet and see element 40 within Figure 2, Abraham)*; receiving responses to said requests from the origin server *(inherent feature in networks yet is also taught within column 2, lines 31-53, Abraham)*; receiving inputs via an interface *(equivalent to network management program; see column 2, lines 30-32, Abraham)* to make a selection for logging of some or all of a plurality of fields that may be present in each of said requests and responses and to make a specification of a sequence in which the selected fields are to appear

in a log file *(see column 14, lines 9-28, Abraham)*; storing in a first data structure, for each selected field, a value indicating a position in the specified sequence of the selected field (*see Hyder below*); in response to receiving each of the request and responses, obtaining information for each selected field associated with the corresponding request or response and storing the information in a second data structure, in a sequence independent of the specified sequence (*see Hyder below*), storing in a third data structure, for each selected field, a reference to the corresponding information stored in the second data structure, including storing, based on the first data structure, each reference in a location of the third data structure that corresponds to the position in the specified sequence of the selected field (*see Hyder below*); and using the third data structure to output the information for each selected field in the second data structure to a log file, such that the information for each selected field appears in the log file according to the specified sequence *(see column 15, lines 34-40, Abraham).*

While Abraham teaches a network management system capable of logging the network data however, Abraham fails to explicitly cite the claimed data structures. In particular, Abraham does not explicitly cite a first data structure to store a value indicating a position in the specified sequence for each selected field; a second data structure to store information corresponding to each selected field; and a third data structure to store a reference to the information stored in the second data structure, including storing the reference in a location of the third data structure that corresponds to the position in the specified

sequence of a field corresponding to the reference. In the same field of endeavor, Hyder teaches a system for maintaining network information. In particular, Hyder teaches a control data structure and packet data structure *(see Figure 4 and column 8, line 35 - column 9, line 12, Hyder).* The control data structure has within it a storage section (data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see column 9, line 2 and Figure 4, Hyder)*. The control data structure also has within it a storage section (data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Finally Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure (deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*. The data from the data structures (including the third data structure) are used for inputting and outputting data held within (or corresponding to) them. The data structures of Hyder's design allow the system to maintain network information (log network data). Therefore it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Abraham with those of Hyder, to teach how data structures can be associated with network information *(see column 5, lines 13-15, Hyder)*.

6. With regards to claim 28, Abraham teaches through Hyder, the method wherein the interface allows creation of new fields in addition to the plurality of fields *(see element 230, Figure 7A, Abraham)*.

7. With regards to claim 29, Abraham teaches through Hyder, the method wherein the information for each field is converted to an ASCII representation and is of variable length (Huckins allows the data to be displayed in ASCII *(equivalent to low-level command; see column 53, lines 15-23, Abraham).*

8. With regards to claim 30, Abraham teaches through Hyder, the method wherein each location in the first data structure is pre-initialized to contain a flag before the specified sequence is stored, the flag to be utilized as an indicator that the corresponding field was not selected for logging *(see column 13, lines 30-33, Abraham).*

9. With regards to claim 31, Abraham teaches through Hyder, the method wherein the second data structure and the third data structure are created to respond to logging for the corresponding request or response and destroyed once logging for the corresponding request or response is completed *(It is inherent that new records (equivalent to the claimed data structures) are created when needed. Abraham teaches the claimed trait within the flowchart presented in figure 7A by element 246).*

10. With regards to claim 32, Abraham teaches through Hyder, the method wherein the first data structure persists through logging for the requests and responses *(Abraham's design allows for records to only be modified if needed; see element 246 of the flowchart within Figure 7A).*

11. With regards to claim 33, Abraham teaches through Hyder, the method wherein using the third data structure to output the information further comprises sequentially accessing the third data structure to read the position of the information corresponding to each selected field and accessing the second data structure to read information corresponding to each selected field at the position indicated by the reference *(see elements 246 and 254 within the flowchart in Figure 7A, Abraham).*

12. With regards to claim 34, Abraham teaches through Hyder, a device for logging information in a network cache, the network cache serving a plurality of clients *(equivalent to devices within the intranet and see element 44 within Figure 2, Abraham)* on behalf of an origin server *(equivalent to devices within the internet and see element 40 within Figure 2, Abraham)*, the device comprising: an interface to allow selection of a protocol, including a selection for logging of some or all of a plurality of fields of a message to be received from anyone of the origin server and the plurality of clients, the fields corresponding to the selected

protocol *(see column 14, lines 9-28, Abraham)*, and a specification of a sequence in which the selected fields are to appear in a log file of the network cache, wherein the interface further allows changing said selections to modify the log file's format while the network cache is running *(see column 7, lines 29-31, Abraham)*; a protocol specific application module to obtain information for each selected field associated with the message *(equivalent to network management program; see column 2, lines 30-32, Abraham)*; a first data structure to store a value indicating a position in the specified sequence for each selected field; a second data structure to store information corresponding to each selected field; a third data structure to store a reference to the information stored in the second data structure, including storing the reference in a location of the third data structure that corresponds to the position in the specified sequence of a field corresponding to the reference; and a protocol independent log module to receive information for each selected field from the protocol specific application module and to store the information for each selected field in the log file in the sequence specified *(see column 15, lines 34-40, Abraham).*

While Abraham teaches a network management system capable of logging the network data however, Abraham fails to explicitly cite the claimed data structures. In particular, Abraham does not explicitly cite a first data structure to store a value indicating a position in the specified sequence for each selected field; a second data structure to store information corresponding to each selected field; and a third data structure to store a reference to the information

stored in the second data structure, including storing the reference in a location of the third data structure that corresponds to the position in the specified sequence of a field corresponding to the reference. In the same field of endeavor, Hyder teaches a system for maintaining network information. In particular, Hyder teaches a control data structure and packet data structure *(see Figure 4 and column 8, line 35 - column 9, line 12, Hyder).* The control data structure has within it a storage section (data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see column 9, line 2 and Figure 4, Hyder)*. The control data structure also has within it a storage section (data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Finally Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure (deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*. The data structures of Hyder's design allow the system to maintain network information (log network data). Therefore it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Abraham with those of Hyder, to teach how data structures can be associated with network information *(see column 5, lines 13-15, Hyder)*.

13. With regards to claim 36, Abraham teaches through Hyder, the device wherein the interface allows creation of new fields in addition to the plurality of fields *(see element 230, Figure 7A, Abraham)*.

14. With regards to claim 38, Abraham teaches through Hyder, the device wherein the interface is a graphical user interface *(see column 14, line 11, Abraham)*.

15. With regards to claim 39, Abraham teaches through Hyder, the device wherein the interface is a command line interface *(equivalent to low level commands; see column 53, lines 15-23, Abraham).*

16. With regards to claim 46, Abraham teaches through Hyder, the device wherein the interface further allows changing said selections to modify the log file's format while the network cache is running *(Within Figure 7A, it is shown how modifications can be made without shutting down the application).*

17. With regards to claim 47, Abraham teaches through Hyder, the method wherein the interface allows changing the selection and the sequence to modify the log file's format while the network cache is running *(see column 7, lines 29-31, Abraham).*

18. With regards to claim 48, Abraham teaches through Hyder, a method of operating a network cache, the method comprising: receiving requests from a plurality of clients *(equivalent to devices within the intranet and see element 44, in Figure 2, Abraham)* for contents maintained by a server *(equivalent to devices within the internet and see element 40 within Figure 2, Abraham)* and responses to the requests from the server *(inherent feature in a network yet is also taught within column 2, lines 31-53, Abraham)*, wherein the requests and responses are formatted according to a protocol; receiving a selection of a first number of fields from a plurality of fields of the protocol that may be present in each of the requests and responses *(see column 14, lines 9-28, Abraham)*, and receiving a specification of a first sequence in which the first number of fields are to appear in a log file of the network cache; recording the first number of fields extracted from a first one of the requests and responses into the log file according to the first sequence *(see column 7, lines 29-31, Abraham)*; storing in a first data structure a value indicating a position in the first sequence of each of the first number of fields; obtaining information of each of the first number of fields associated with the first one of the requests and responses and storing the information in a second data structure, in a sequence independent of the first sequence, storing in a third data structure a reference to the information for each of the first number of fields stored in the second data structure, including storing each reference in a location of the third data structure that corresponds to the position in the first sequence of the fields corresponding to the reference; and

using the third data structure to output the information for each of the first number of fields in the second data structure to the log field, such that the information for each of the first number of fields appears in the log file according to the first sequence; while operating the network cache, receiving inputs that changes selection of fields from the first number of fields to a second number of fields and specifies a second sequence; and recording the second number of fields extracted from a second one of the requests and responses into the log file according to the second sequence *(see column 15, lines 31-40, Abraham)*.

While Abraham teaches a network management system capable of logging the network data however, Abraham fails to explicitly cite the claimed data structures. In particular, Abraham does not explicitly cite a first data structure to store a value indicating a position in the specified sequence for each selected field; a second data structure to store information corresponding to each selected field; and a third data structure to store a reference to the information stored in the second data structure, including storing the reference in a location of the third data structure that corresponds to the position in the specified sequence of a field corresponding to the reference. In the same field of endeavor, Hyder teaches a system for maintaining network information. In particular, Hyder teaches a control data structure and packet data structure *(see Figure 4 and column 8, line 35 - column 9, line 12, Hyder).* The control data structure has within it a storage section (data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see*

column 9, line 2 and Figure 4, Hyder). The control data structure also has within it a storage section (data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Finally Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure (deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*. The data from the data structures (including the third data structure) are used for inputting and outputting data held within (or corresponding to) them. The data structures of Hyder's design allow the system to maintain network information (log network data). Therefore it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Abraham with those of Hyder, to teach how data structures can be associated with network information *(see column 5, lines 13-15, Hyder)*.

19. With regards to claim 49, Abraham teaches through Hyder, the method wherein recording the first number of fields extracted from a first one of the requests and responses into the log file according to the first sequence comprises: storing in a first data structure a value indicating the position in the first sequence of each of the first number of fields; obtaining information for each of the first number of fields associated with the first one of requests and responses and storing the information in a second data structure, in a sequence independent of the first sequence, storing in a third data structure, based on the first data structure, a

reference to the information for each of the first number of fields stored in the second data structure, including storing each reference in a location of the third data structure that corresponds to the position in the first sequence of the corresponding field; and using the third data structure to output the information for each of the first number of fields in the second data structure to the log file, such that the information for each of the first number of fields appears in the log file according to the first sequence *(see elements 246 and 254 within the flowchart in Figure 7A, Abraham).*

20. The obviousness motivation applied to claims 21, 27, 34 and 48 are applicable to their respective dependent claims.

Response to Arguments

The amendment received on October 9, 2008 has been carefully examined but is not deemed fully persuasive. The following are the examiner's response to the applicant's arguments.

The first point of contention addressed by the applicant concerns the claimed interface to make selection for logging of some or all of a plurality of fields of the protocol. Applicant contends that while the Abraham's prior art does teach allowing an administrator to make the selection of a protocol for logging; it is silent with respect to teaching a user interface for allowing an administrator to select some or all of a plurality of fields of the protocol for logging. The examiner respectfully disagrees with this

assertion. The claim language cites "to select <u>some</u> <u>or</u> all of a plurality of fields of the protocol for logging." Hence, if one item related to a protocol is elected for logging, it satisfies the claim requirements. Abrahams teaches the logging of at least one item related to a selected protocol hence it satisfies the claim requirement; see column 14, lines 9-28, Abraham.

The second point of contention addressed by the applicant concerns the claimed first, second and third data structures. The applicant contends that neither prior arts teach such a claim limitation. The examiner respectfully disagrees. The applicant is reminded that a data structure itself is merely a data holder (whether it is a table, a variable or other forms of data structures such as classes and linked lists). A data structure can even exist within another data structure (for instance a variable within a larger data structure is still a data structure). It is inherent that data structures exist within any digital design (that includes network designs such as Abraham's). In the same field of endeavor, Hyder teaches a system for maintaining network information. In particular, Hyder teaches a control data structure and packet data structure *(see Figure 4 and column 8, line 35 - column 9, line 12, Hyder).* The control data structure has within it a storage section (a data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see column 9, line 2 and Figure 4, Hyder)*. The control data structure also has within it a storage section (a data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Finally Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure

(deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*. The data from the data structures (including the third data structure) are used for inputting and outputting data held within (or corresponding to) them. The data structures of Hyder's design allow the system to maintain network information (log network data). Applicant further contends that the first data structure needs to be enabled to store for each of the plurality of fields. However as explained above, the claims only require one item related to a selected protocol since the language "some" is used. Thus, Abrahams teaches the logging of at least one item related to a selected protocol hence it satisfies the claim requirement; see column 14, lines 9-28, Abraham. And Hyder teaches a control data structure that has within it a storage section (a data structure), for storing numeric offset (deemed equivalent to the claimed first data structure storing position data) *(see column 9, line 2 and Figure 4, Hyder)*. The applicant then further contends that the third data structure stores a reference to the data of each selected field in the second data structure. The examiner is aware of the applicant's intent and believes that he has provided sufficient evidence within the art of record. A pointer in the computing world is a reference. Hyder's packet data structure has a pointer *(pointer is a reference)* to the control data structure (deemed equivalent to the claimed third data structure) *(see column 8, lines 35-36 and Figure 4, Hyder)*.

The third point of contention addressed by the applicant concerns the claim feature of "storing the information in the second data structure in a sequence independent of the specified sequence." The applicant contends that neither prior art teach such a claim limitation, the examiner respectfully disagrees. For the second data

structure the examiner has relied on the free form data within the storage section. The control data structure of Hyder's disclosure has within it a storage section (a data structure), for storing free form data (deemed equivalent to the claimed second data structure) *(see column 9, lines 6-12, and Figure 4, Hyder)*. Free form data is not in any particular sequence. Nor is any mention made of any dependence on sequence within Hyder with respect to the free form data. Hence, free form data within the storage section is equivalent to the claimed second data structure in a sequence independent of the specified sequence.

The fourth and final point of contention addressed by the applicant concerns the combination of the prior arts of record. In particular the applicant contends that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the data from the data structures (including the third data structure) are used for inputting and outputting data held within (or corresponding to) them. The data structures of Hyder's design allow the system to maintain network information (log network data). Therefore it would have been obvious to one skilled in the art, during the time of the invention, to have combined the teachings of Abraham

with those of Hyder, to teach how data structures can be associated with network information *(see column 5, lines 13-15, Hyder)*.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to AZIZUL CHOUDHURY whose telephone number is (571)272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton B. Burgess can be reached on (571) 272-3949. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Patrice Winder/
Primary Examiner, Art Unit 2445

/A. C./
Examiner, Art Unit 2445